ITEA Journal 2011; 32: 95–102

DCF®—A JAUS and TENA Compliant Agent-Based Framework for Test and Evaluation of Unmanned Vehicles

Nicholas Lenzi, Benjamin Bachrach, Ph.D., and Vikram Manikonda, Ph.D.
Intelligent Automation, Inc., Rockville, MD

Real-world applications for unmanned and autonomous systems (UAS) teams continue to grow, and the scale and complexity of the teams are continually increasing. To reduce life cycle costs and improve test and evaluation (T&E), we increasingly need to develop a generalized framework that can support the design and development of T&E approaches for multi-UAS teams and validate the feasibility of the concepts, architectures, and algorithms. This challenge is most significant in the cognitive–social domains, where the development of test approaches and methodologies are difficult because of the emergent nature of behaviors in response to dynamic changes in the battlespace. Today much of the initial validation effort is done using simulations, which unfortunately rarely capture the complexity of real world effects related to net-centric communications, vehicle dynamics, distributed sensors, physical environment (terrain), external disturbances, etc. Furthermore, very often high fidelity simulations do not scale because the number of UAS increases. This article addresses DCF®—a JAUS and TENA compliant agent-based T&E framework for simulated, mixed-model (virtual and live–hardware in the loop) and live testing of teams of unmanned autonomous systems.

Key words: DCF®; emergent behavior; real world battlespace; simulation; UAV teams; unmanned autonomous systems; validation.

The successful deployment of unmanned platforms in the battlefield has led to an increased demand for greater numbers of unmanned and autonomous systems (UAS). Coupled to this increase in demand is the expectation of greater levels of autonomy for these systems (DOD, 2009). There is a compelling need for the development of flexible test and evaluation (T&E) frameworks that can address the challenges associated with testing increasingly complex systems over shorter testing cycles (DOD, 2009; Streilein, 2009).

Under an ongoing effort with the Test Resource Management Center, Unmanned and Autonomous System Test program, we have developed an integrated agent-based T&E framework for Simulated, Mixed-Model, and Live Test and Evaluation of teams of unmanned autonomous systems. At the core of this T&E architecture is an agent-based distributed control framework (DCF) (Kulis et al. 2008; Manikonda et al 2008a; 2008b). As part of ongoing efforts, an enhanced JAUS and TENA compliant version of DCF is being developed and tested. A user interface with integrated T&E environment development, simulation, and command and control capabilities has been implemented. The DCF is also being tested and evaluated at the Armament Research Development and Engineering Center (ARDEC) in Picatinny Arsenal in a relevant test environment.

In this article, we discuss the details of these recent enhancements and present initial results from a technology development conducted at ARDEC, including a discussion of the features of the vignette editor, our implementation of JAUS and TENA compliance, the details of the technology demonstration, our conclusions, and future research directions.

Vignette editor

DCF® (IAI, Rockville, MD) is a small and lightweight T&E framework that may be deployed on any computing architecture that supports the Java Virtual Machine (Kulis et al. 2008; Manikonda et al. 2008a; 2008b). The DCF adopts an agent-based modeling paradigm and simplifies the implementation and test of distributed algorithms, supports mixing of virtual robot agents with real robot agents, and enables

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **MAR 2011**	2. REPORT TYPE	3. DATES COVERED **00-00-2011 to 00-00-2011**

4. TITLE AND SUBTITLE **DCF-A JAUS and TENA Compliant Agent-Based Framework for Test and Evaluation of Unmanned Vehicles**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Intelligent Automation, Inc,15400 Calhoun Drive, Suite 400,Rockville,MD,20855**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**8**	

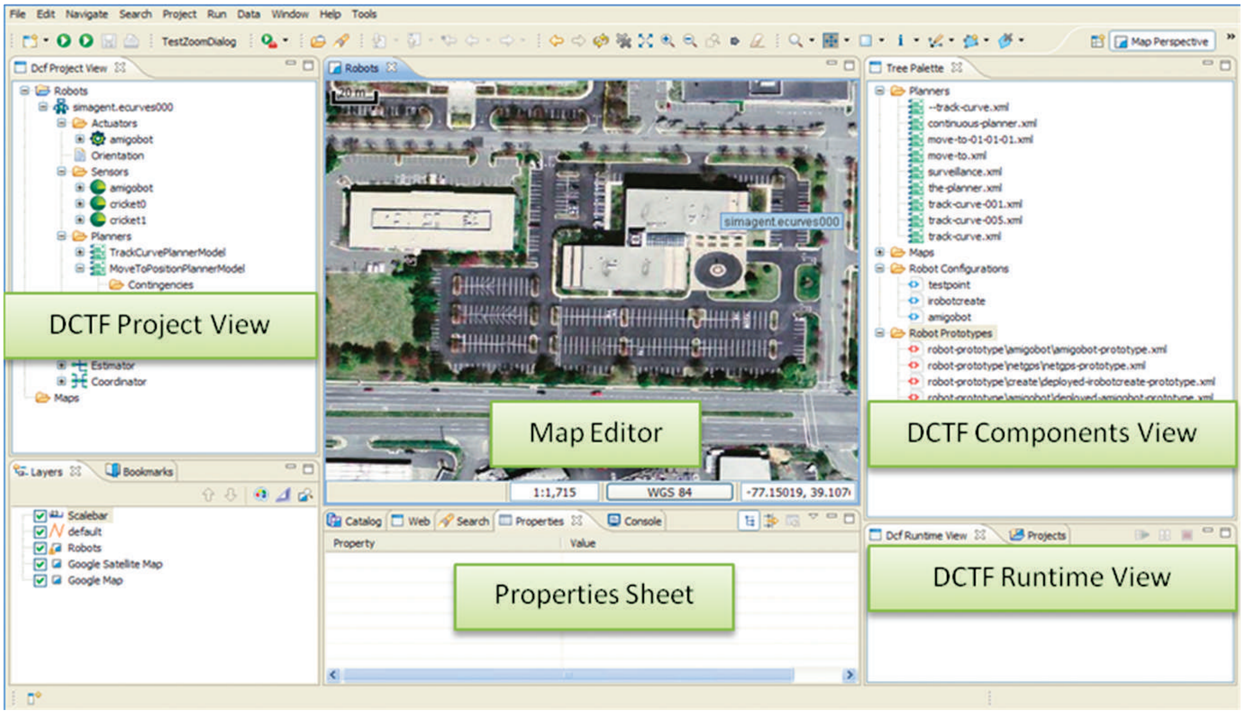

Figure 1. DCF vignette editor.

data sharing via peer-to-peer messaging. A controls-centric design is adopted wherein a robot agent is composed of a sensor layer, state estimators, motion planners, and an actuation layer. Algorithms are implemented as plug-ins and include hardware device abstraction and self-contained sensor–actuator drivers, with components loaded at run time via XML configuration files. The DCF currently provides drivers for a variety of robots (e.g., iRobot Creates, Pioneers, Amigobots, FireAnt, LAGR), and a wide ranges of sensors (e.g., digital encoders, sonars, stereo cameras, GPS receivers, inertial navigation systems, LIDARs, and cameras). The DCF also provides hardware-in-the-loop support, discrete-time and real-time simulations, built-in equation solvers, distribution across multiple computing resources with repeatable results, cross-layer (network-level) modeling, and human in the loop support.

To further facilitate T&E, the DCF now provides a user interface called the *vignette editor (Figure 1)*. The functions of the vignette editor include visualizing the state of the UAS team, creating T&E scenarios, monitoring the UAS team performance, and generating automated T&E reports. Most importantly, the vignette editor provides the user with the ability to issue real-time commands to the team and to upload a new distributed control algorithm (mission) on the fly. A description of the main components of the vignette editor follows.

Project View

The project view provides a tree view of all of the robots. From this view, robots can be added and configured and missions can be built and assigned. Each robot element displays the set of a*ctuators*, *sensors, state estimators, coordinators,* and *planners* based on the robot agent architecture.

Run-time View

The run-time view provides a view of all of the *robot agents* in the DCF community. From this view, the contents of the sensor map and the active plan are displayed. Each element within the tree can be selected, and the corresponding parameters such as serial port or desired position can be viewed via the properties sheet view. The run-time view can be extended to support new device types as programmers create them. Via the context menu and the local toolbar, users can assign, pause, and resume plans; display a live video feed; and run the robot by remote control.

Components View

The components view provides a view of all components that can be added. The user simply has to drag the selected component onto the map to make the necessary change.

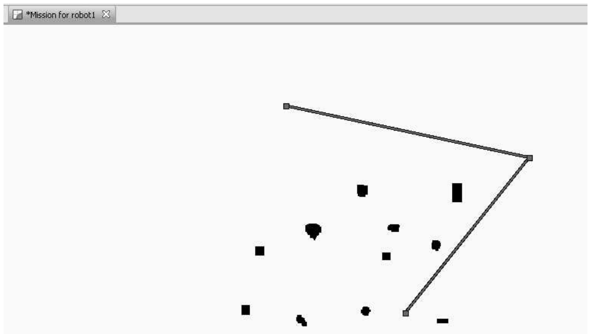

Figure 2. Serial mission editor with selected waypoints.

Mission Building Editors

In addition to building missions with XML files, two graphical mission editors are supported. The serial mission editor allows the user to string discrete behaviors into a serial sequence. These behaviors are then executed by the robot sequentially *(Figure 2)*. A state machine mission editor allows mission builders to use *behaviors* from the component view and visually configure them in a finite state machine *(Figure 3)*.

Map Editor

The map editor is an eclipse editor implemented within the uDig[1] application to provide a way to display a series of map layers such as geographical maps and road locations. It provides a two-dimensional canvas to display layers on a map such as robot locations. Users can drag robots to adjust their positions and orientations.

Web Map Tile Server Visualization

A series of uDig renderers, using any standard Web

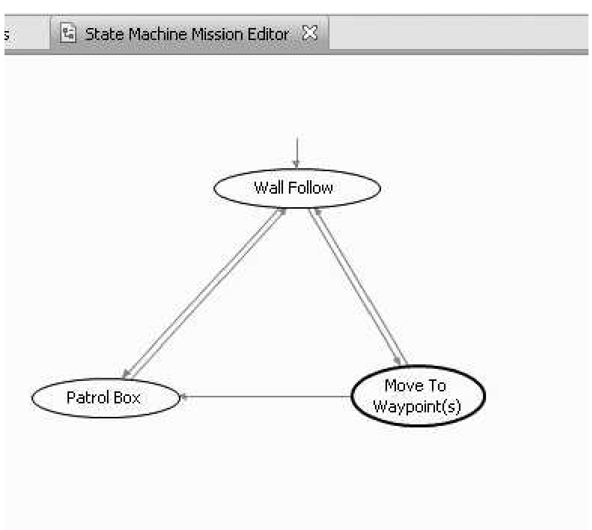

Figure 3. State machine editor.

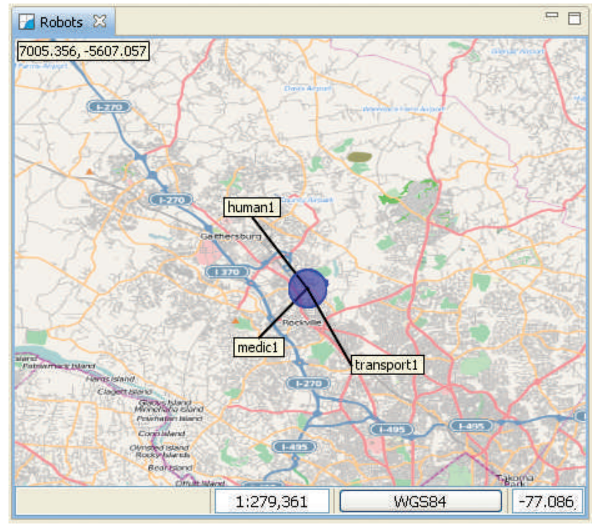

Figure 4. DCF map view with open street maps.

map tile server for the back end, were created to visualize street maps, aerial photographs, and terrain maps. These components download the images from Web servers, cache them locally, and display them accordingly in the map editor *(Figure 4)*.

Digital Terrain Elevation Data (DTED) Visualization

DTED is a file format used by the military to encode elevation data over a large scale. Leveraging an existing technology, a DTED-based UDig renderer was built. The DTED renderer displays a topographic map built from DTED level 0, 1, or 2 *(Figure 5)*.

Streaming Video

The vignette editor supports any number of incoming video streams, as long as their sources are known. Users can right click a robot in the run-time view, and select Show Video Stream to bring up the video window for a particular robot.

Metrics Evaluation Integration

Visualization capabilities for real-time metrics have been incorporated into the vignette editor. Users may visualize metrics via configurable plots during run time using the JFreeChart library. Numerous types of plots are supported via JFreeChart. An example of a time sequence chart showing the navigation error of a robot is shown in *Figure 6*.

Joint Architecture for Unmanned Systems (JAUS) Compliance

JAUS is a messaging standard that has been mandated by the DoD to facilitate interoperability between unmanned systems (OpenJAUS 2010). In

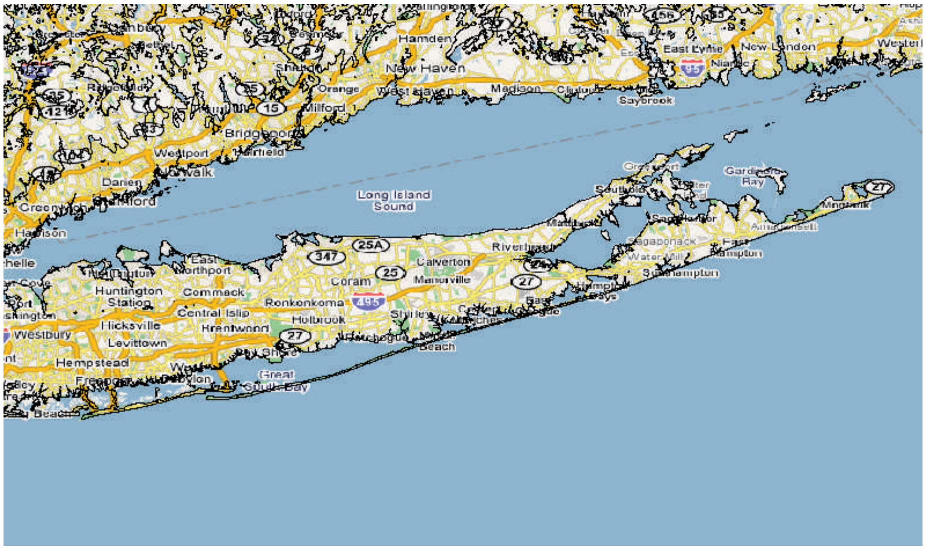

Figure 5. Screenshot Google map with DTED topographic overlay.

addition to the messaging standard, JAUS also defines a series of hierarchically organized software-naming schemes to reduce confusion. These object names are *Subsystem, Node, Component, and Instance*.

A subsystem is generally viewed as a complete hardware and software solution such as an unmanned ground vehicle (UGV) platform or an operator control unit. A node is generally viewed as a process running on a dedicated central processing unit. Components are logically organized software components that generally perform some specific sensing or driver level task within the node.

There are three levels of JAUS compliance—Level 1, Level 2, and Level 3. Level 1 compliance indicates that all communication between JAUS subsystems is done via JAUS messages. Level 2 compliance indicates that all communication between JAUS nodes is done via JAUS messages. Level 3 compliance indicates that all communication between JAUS components is done via JAUS messages.

To enable JAUS compliance, we implemented a DCF-style JAUS controller that sends JAUS messages to specific JAUS components on the platform. Our initial implementation is Level 1 compliant because it sends and receives messages at the subsystem level. The JAUS controller was designed to directly interface with the Primitive Driver, Reflexive Driver, Local Waypoint Driver, and Global Waypoint Driver to support driving the platform. Additionally, periodic updates of important sensor data were required. Global Pose Sensor and Local Pose Sensor were implemented to support the creation of higher-level DCF behaviors that allowed more complex tasks such as perimeter surveillance.

Other data that were implemented included the image data from the visual sensor and platform operation data from the primitive driver. Additionally, FireAnt-specific experimental custom messages were implemented to support control of the pan/tilt/zoom camera, querying the encoders, and querying the LIDAR.

TENA Compliance

The Test and Training Enabling Architecture (TENA) is a middleware designed to support interaction between remote software components (DOD CTEIP 2010a). The specific applications that use TENA are T&E applications where users want to integrate data collected with remote test ranges.

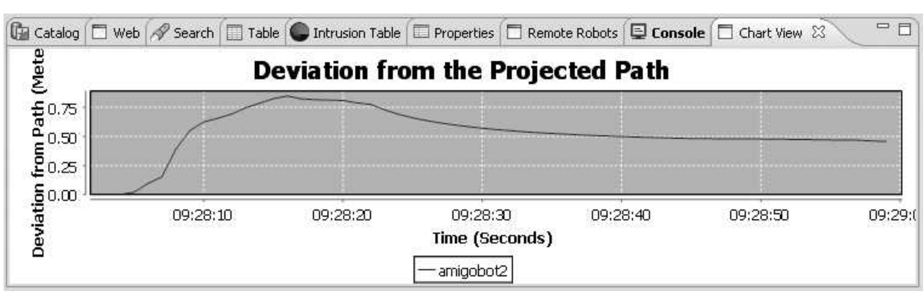

Figure 6. Real-time metric display.

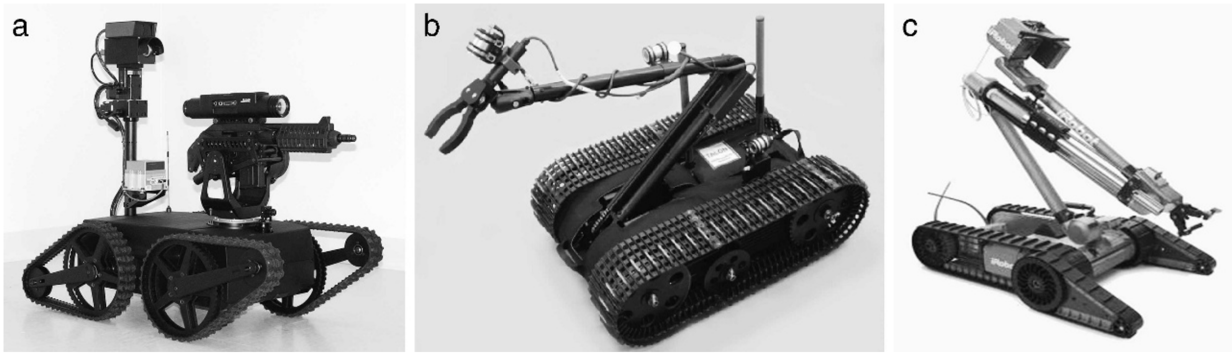

Figure 7. Some of ARDEC UGV platforms: (a) FireAnt, (b) Talon, and (c) Packbot.

TENA classes are implemented in their own programming language, which is similar in syntax to C++, and compiled remotely by the TENA community at the TENA Software Development Architecture Web site.

To facilitate interfacing with TENA, we developed a DCF-TENA integration approach to support relaying DCF robot agent data across the TENA infrastructure and support waypoint tasking from remote TENA applications. Our initial implementation adopts the "gateway" architecture as discussed in DOD CTEIP (2010b). A series of TENA classes was developed and compiled, which are available via the tena-sda Web site (DOD CTEIP 2010a). TENA application programmers can task robots to an (X, Y) location using the TENA method moveToLocation. Additionally, they can query the position of the robots as well. Future development will be done by integrating with an existing TENA repository, and logging data to it.

Evaluation of DCF at ARDEC in Picatinny Arsenal

ARDEC personnel at Picatinny Arsenal have developed the Firestorm system, a fully integrated and scalable decision support tool suite for the mounted–dismounted warfighter–commander. Firestorm is an open, extensible, and scalable family of tools that support network centric warfare and can be configured for user experimentation in either a virtual or field environment. ARDEC is also developing the concept of a Joint Manned–Unmanned System Team (JMUST), for which target handoff and sharing of situational awareness data between humans and unmanned and autonomous systems working together have been demonstrated. This is a groundbreaking program in terms of implementation of advanced concepts for human–UAS teaming in combat operations. Some examples of unmanned systems currently being integrated at ARDEC include military robots such as the FireAnt, PackBot, Talons, and Scouts *(Figure 7).* However, because new unmanned platforms (manufactured by different vendors with different levels of JAUS compliance, if at all) are being integrated into Firestorm, new challenges are emerging. There is a critical need at ARDEC for a framework to coordinate the behavior of these platforms and to test the performance of teams of unmanned systems.

To address this need, ongoing efforts with ARDEC are designed to validate and demonstrate the benefits of DCF to a multirobot coordination task performed in a realistic environment. Use cases for "perimeter surveillance" scenarios were identified and implemented in two stages.

Single UGV perimeter surveillance. For an unmanned system to conduct autonomous perimeter surveillance, the operator would provide a region (such as a building, for example) around which the unmanned system should conduct surveillance. The unmanned platform would have to generate a surveillance path plan around the region of interest as a sequence of waypoints. At each way point the unmanned system would conduct a surveillance activity, such as searching for candidate targets using a camera (the target could be predefined or a moving object). If such a target is identified, a message would be sent to the operator control unit together with an image of the target.

Multiple UGV perimeter surveillance. As in the previous discussion, the operator would provide a region around which the team of unmanned systems should conduct surveillance. The unmanned platforms would collaboratively generate a surveillance path plan around the region of interest as a sequence of waypoints for each of the unmanned platforms. At each waypoint the unmanned systems would conduct a surveillance activity, as in the single platform case. Higher and more interesting behaviors can be achieved by multi-UAS systems. For example, if a given target is identified by UAS-1 (say UGV 1), it could be confirmed or handed over to UAS-2. UAS-2 could be commanded to follow the target while UAS-1 continues the surveillance, etc. This decision could be taken with or without operator intervention.

The main stages of the perimeter surveillance mission implemented in collaboration with ARDEC personnel were the following:

a. **Selection of region of interest.** The operator uses the mouse to indicate on the vignette editor the region over which the UGVs are to perform surveillance *(Figure 8)*. This region can be any nonintersecting polygon. Once the region is selected, the planner defines a sequence of waypoints that the UGVs are to traverse during the surveillance.
b. **Deployment of UGV team.** The UGV robot agents negotiate over equally spaced starting positions of each platform. Once a starting point for each platform is assigned, both platforms navigate to their respective starting positions *(Figure 9)*.
c. **Surveillance.** Both robots start a clockwise rotation pattern traversing each of the waypoints in the perimeter of the region of interest. At each waypoint the platforms stop and conduct a target detection search where they pan their cameras toward the outside of the region of interest *(Figure 10)*.
d. **Target detection.** If a target is detected during any point of the surveillance mission, a pop-up menu is displayed on the vignette editor prompting the operator to take an action such as "*Continue Surveillance*," "*Remote Control*," "*Follow Target*," etc. If no action is taken by the operator within a timeout period, the surveillance mission continues.
e. **End of mission.** At the completion of the mission, the operator has the choice of manually teleoperating each of the platforms or giving them all a command to go back to their starting positions.

The use cases listed were implemented over a number of visits to Picatinny Arsenal. In June 2010, a coordinated perimeter surveillance mission using two FireAnt UGVs was successfully demonstrated. *Figures 11* and *12* show the two platforms performing the prescribed mission.

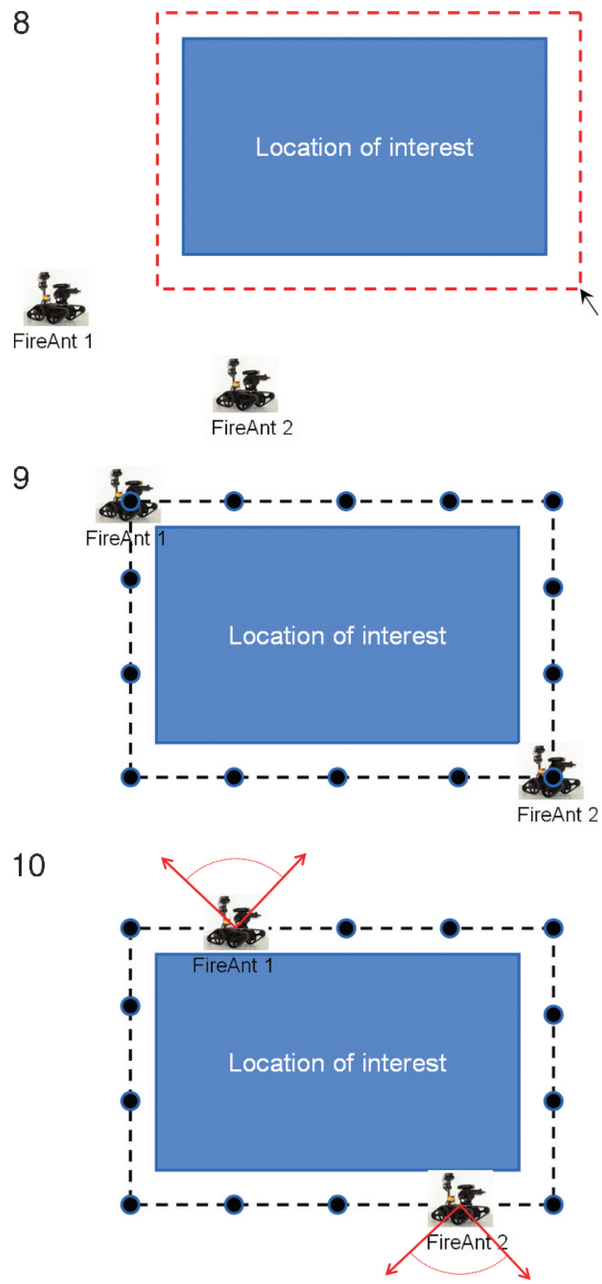

Figure 8. Selection of region of interest.
Figure 9. Deployment of UGV team.
Figure 10. Surveillance.

Conclusions and Future Research

In this article we presented the details of enhancements made to DCF, a T&E infrastructure developed for unmanned and autonomous systems. We focused on the vignette editor, JAUS and TENA compliance, and test and validation at ARDEC. This enhanced system has already reached a technology readiness level of 6 and is currently in the process of being evaluated at Picatinny Arsenal. Future research includes further enhancements to the vignette editor, adding further JAUS compliance levels and improving the initial implementation of the TENA-DCF gateway. Further enhancements to the vignette editor to allow test engineers to develop T&E scripts for command and control approaches for navigation through a cluttered terrain and urban terrain, using non–line-of-sight instrumentation techniques; UAS team coordination and performance in nondeterministic, unscripted

Figure 11. FireAnt 1 and FireAnt 2 during coordinated perimeter surveillance mission.

modes of operation (emergent behavior); and fault tolerance under various failure modes and bandwidth constraints (hardware, sensor, network) need to be developed. While DCF currently supports drivers for a large class of UGVs, in future work UGV drivers and sensor payload for Department of Defense UGVs will be further developed and tested. ❑

NICHOLAS LENZI is a software engineer at Intelligent Automation, Inc. He received his bachelor of science degree in computer science from the University of Maryland, Baltimore County, in 2006. His interests include control, robotics, distributed applications, multimedia applications, databases, design patterns, and software engineering. Mr. Lenzi is the lead engineer on the DCF effort. E-mail: nlenzi@i-a-i.com

DR. BENJAMIN BACHRACH is the vice president of the Sensors, Signals and Systems Group and acting director of the Robotics Group at Intelligent Automation Inc. He received his bachelor of science degree in electrical engineering from Tel Aviv University, Israel, in 1987. He continued his education at the University of Maryland, at College Park, where he earned his master of science and his doctor of philosophy degrees in electrical engineering, with specialization in the area of control theory and communications in 1992 and 1997, respectively. Since then, Dr. Bachrach has been involved in a variety of projects in diverse areas. Some of these projects include 3D-based systems for automated evidence comparison, robotic systems for physical rehabilitation, vision-based target detection and tracking, beam stabilization for laser designators, and control of teams of unmanned systems, among others. E-mail: bach@i-a-i.com

DR. VIKRAM MANIKONDA is the president of Intelligent Automation Inc. He received his bachelor of engineering degree in electrical engineering from the Birla Institute of Technology and Science, India, in 1992; his master of science and his doctor of philosophy degrees, both in electrical engineering, from the University of Maryland at College Park, in 1994 and 1997, respectively. His research interests include intelligent control, robotics, motion

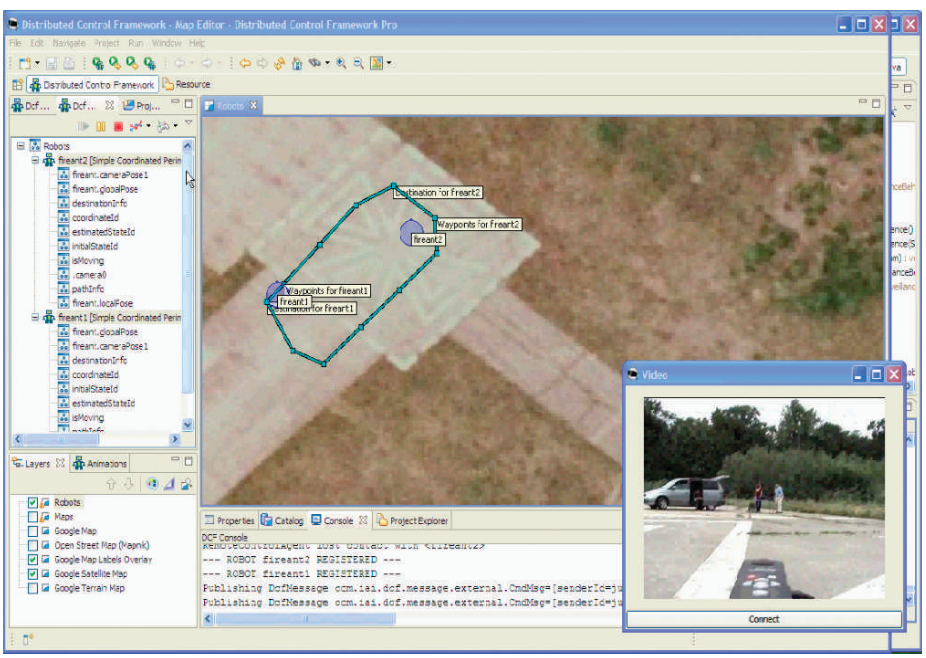

Figure 12. IAI's vignette editor operating as operator control unit to control a coordinate perimeter surveillance mission at Picatinny Arsenal. The light blue disks correspond to the UGVs (Fireant 1 and Fireant 2); the perimeter under surveillance is shown as a sequence of waypoints that the UGVs are to follow. The bottom right corner shows a live video of Fireant 2 taken in real time.

description languages, multiagent systems for modeling and simulation, and air traffic control and management. Dr. Manikonda was the principal investigator on the initial TRMC BAA supporting this effort. E-mail: vikram@i-a-i.com

Endnotes

[1]uDig is a GIS framework for Eclipse (see http://udig.refractions.net/).

References

DOD. 2009. *Office of the Secretary of Defense FY2009-2034 unmanned systems integrated roadmap*, 2nd ed. http://www.jointrobotics.com/documents/library/UMS%20Integrated%20Roadmap%202009.pdf (accessed November 24, 2010).

DOD CTEIP. 2010a. Washington, D.C.: Central Test and Evaluation Investment Program (CTEIP). www.tena-sda.org. (accessed November 24, 2010).

DOD CTEIP. 2010b. *The Test and Training Enabling Architecture.* Architecture Reference Document, version 2002. Washington, D.C.: Central Test and Evaluation Investment Program (CTEIP). https://www.tenasda.org/download/attachments/6901/TENA-ArchitectureReferenceDocument-002.pdf?version=1&modificationDate=1142526068000 (accessed November 24, 2010).

Kulis, Z., V. Manikonda, B. Azimi-Sadjadi, and P. Ranjan. 2008. The distributed control framework: a software infrastructure for agent-based distributed control and robotics, presented at the American Control Conference, Seattle, WA, 2008.

Manikonda, V., Z. Kulis, and Nick Lenzi. 2008a. A motion description language and its applications to test, evaluation and control of unmanned autonomous system teams. Annual ITEA Technology Review, Colorado Springs, CO, 2008 (presentation only)

Manikonda, V., P. Ranjan, Z. Kulis, and B. Azimi-Sadjadi. 2008b. An agent-based distributed control framework for design, simulation and deployment of unmanned vehicles teams. In *Proceedings of the Simulation Interoperability Workshop*. Providence, RI: Intelligent Automation, Inc.

OpenJAUS, 2010. OpenJAUS Home. http://www.openjaus.com/ (accessed November 24, 2010)

Streilein, James, J., 2009. Test and evaluation of highly complex systems, *ITEA Journal* 30 (1), 3–6.

Unmanned and Autonomous Systems Testing Roadmap, March 3, 2009.

Acknowledgments

The work reported in this paper was funded in part under contracts W9124Q-07-C-0685 and W9124Q-09-P-0272 from the Test Resource Management Center under the Unmanned and Autonomous System Testing program.

15th Annual ITEA Test Instrumentation Workshop

TEST AND TRAINING: Converging Solutions in a New Millennium

MAY 9–12, 2011

Tuscany Suites and Casino
Las Vegas, Nevada

TECHNICAL CHAIR
Mr. Cameron Bruce
1 Administration Circle, MS 1106
China Lake, CA 93555-6100
760-939-0307
cameron.bruce@navy.mil

REGISTRAR
Mrs. Jean Shivar
703.631.6225 • jean@itea.org

Workshop Focus

Providing innovative test instrumentation solutions to a continually evolving set of challenges is vital to meeting the warfighting technology and development needs of the future.

This workshop, located near the largest concentration of test and training ranges in the world, will explore the innovative test and training instrumentation solutions to the challenges presented in the complex test environments of today and tomorrow.

Topics that will provide innovative test instrumentation solutions:

- Thematic
- Innovation
- Process
- Programs

Exhibit and Sponsorship Opportunities Available

For more information on the levels and benefits of sponsoring or to exhibit your products and services, please contact Mr. Bill Dallas, 703.631.6226 or wdallas@itea.org.

Hotel Information

Tuscany Suites and Casino
255 E. Flamingo Road
Las Vegas, NV 89169
702.893.8933 • 877-887-2261
www.tuscanylv.com
Reserve your room by April 18 to receive the ITEA rate of $95.

Please check the ITEA website for more details on the program, tutorials being offered,the golf tournament being planned, and much more!

www.itea.org

Hosted by the Antelope Valley, China Lake and Southern Nevada Chapters